Research Article

Optimization of a Semiflexible Demand-Responsive Feeder System in Suburban Areas Using a Memetic Algorithm

Fabio Sartori Vieira,1 Kenneth Sorensen,2 and Pieter Vansteenwegen1

1KU Leuven, KU Leuven Institute for Mobility CIB, Leuven, Belgium
2ANT/OR University of Antwerp Operations Research Group, Antwerp, Belgium

Correspondence should be addressed to Fabio Sartori Vieira; fabio.vieira@kuleuven.be

Received 6 September 2022; Revised 6 January 2023; Accepted 18 January 2023; Published 15 February 2023

Academic Editor: Zhihong Yao

Copyright © 2023 Fabio Sartori Vieira et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traditional bus operations in suburban areas are inefficient due to their fixed routes and timetables. Since suburban operations deal with low demand spread in a large area, the service stays underused for most off-peak hours. In order to render the operation profitable and increase the number of passengers on each bus, operators reduce the frequency of the service, which results in an increase of passenger waiting time for the service. As a solution to this problem, this paper introduces a demand-responsive public bus system that aims to adjust routes and timetables of a semiflexible system to the demand for transportation. The operation still offers a reliable service like the traditional system but aims to reduce the passenger travel time. A memetic algorithm is developed to optimize this demand-responsive system. For a network with 25 bus stops served hourly by three lines and with an average demand of 20 requests per hour, the memetic algorithm is demonstrated to reduce the passenger waiting time with almost 50% in comparison with a traditional system operating in the same network with fixed routes and timetable.

1. Introduction

The growth of (sub)urban areas around big cities, such as new neighborhoods or settlements that operate as commuting towns, puts pressure on public transportation systems. These new urban areas generally have a low population density spread over a large area. Direct connections or extensions of the current routes of the public transport system potentially lead to a nonprofitable service. In order to improve the public transportation system in these areas, the most applied solution is the implementation of feeder lines. Feeder lines zigzag through the suburban area to bring passengers to a hub station where a high-frequency service offers a direct and fast connection to the demand centers (city center, industrial zones, and schools).

Feeder lines partially solve the efficiency problem in suburban areas with low demand, especially during peak hours. However, when demand is lower during off-peak hours, feeder lines with fixed routes and timetables are no longer efficient. Their service is typically optimized for the demand predicted based on historical data, without considering external factors that might influence the daily demand using the service, such as weather conditions or nonregular events.

Another approach to address this type of demand are fully flexible demand-responsive (taxi) services, which operate without using fixed routes or timetables but pick up and drop off passengers according to their needs in a service complimentary to the traditional transit system, typically in a door-to-door service. In the OR literature, such systems are often modeled as a variant of the Dial-a-Ride Problem (DARP). Despite being a very responsive service, the operation of such systems in suburban areas with low demand can also lead to low seat occupancy, and consequently, an expensive operation.

This paper aims to investigate the applicability of a semiflexible demand-responsive transportation system
replacing a traditional feeder system in suburban areas. The fact that this system is semiflexible is a key novelty. This means, on the one hand, it has some characteristics of traditional systems, i.e., a regular frequency of operation, predefined standard routes, and a stop-based service. This “backbone” makes this system recognizable for the users and lowers the threshold for using it. Thus, this system still serves the demand by relying on the structure of the traditional feeder system. On the other hand, it is a demand-responsive system, allowing modifications to the standard routes and adjusting the timetable of each trip while respecting the initial frequency. Therefore, our proposed system falls in the category of a semiflexible demand-responsive feeder systems (DRFS) and intends to offer an optimized service with some level of responsiveness to the actual demand for transportation yet still based on a service running regularly. It should also be noted that the same resources are being used as in the traditional system, so the operator costs remain the same and the focus will be on offering a better service, with roughly the same operational costs.

As the traditional system, our semiflexible DRFS is stop-based. It means that passengers willing to submit a request must indicate beforehand their origin bus stop and walk to the stop at the preferred departure time. Requests for a trip can be submitted until just before the operation starts. Based on a list with all passenger requests, the routes and departure times for the buses operating each line are optimized for each hour of operation of the services. The objective when optimizing the operations of this DRFS is to minimize the sum of the travel times for all passengers, from the moment they arrive at the bus stop until they arrive at the hub station. Therefore, a memetic algorithm (MA) is introduced to optimize the performance of the DRFS from the users’ perspective.

The main contributions of this paper are that we introduce and analyze a new, semiflexible, demand-responsive feeder system, designed for low demand suburban areas. Different from the other DRT systems, our semiflexible system tries to combine the benefits of a fixed system and a fully flexible DRT. Furthermore, a memetic algorithm is designed for optimizing the performance of this new system when serving a set of registered requests. After evaluating the performance of the algorithm, the performance of the system is analyzed and compared to the performance of a traditional feeder system. This allows determining under which circumstances this new system should be considered in practice.

The related literature on public bus transportation systems is discussed in the next section, both for feeder systems in general and for the DRFS. The assumptions for the DRFS proposed in this paper are described in Section 3. The memetic algorithm is discussed in detail in Section 4. Section 5 presents the results of the experiments. First, the efficiency of the memetic algorithm is evaluated by comparing its results with an optimal solution obtained through a mathematical model. Then, the DRFS is compared to the TFS. Several instances are generated, varying the list of requests in each instance. Five different experiments are performed for these instances. One last experiment considers a large network and instances with more requests. Section 6 concludes the paper and discusses opportunities for further research.

### 2. Literature Review

This section first discusses the uprise of demand-responsive systems, with different levels of flexibility. Then, the DRFS is positioned in the state of the art and various closely related systems are discussed.

As an alternative for traditional systems, demand-responsive systems emerged in the last decades and their popularity has been rising over the years. One of the initial, yet still popular, demand-responsive systems is a door-to-door dial-a-ride (DAR) system with the purpose of transporting reduced-mobility or elderly passengers, typically running in parallel to a fixed service [1]. Such a system is also used as a transportation mode for rural communities without scheduled bus services. This fully flexible system is typically operated based on reservations. After the surge of more innovative solutions and intelligent transportation systems (ITS), more demand-responsive transport systems were developed for a wider variety of purposes, complementing the fixed transportation systems [2–5].

Several papers explore case studies to improve the current transit operation systems in rural areas (i.e., see [6]), the implementation of alternative DRT systems and other modes of operation (i.e., see [7]), or the optimal redesign of the DRT systems to increase efficiency [8]. The common characteristic identified in transit operation in suburban or rural areas is the necessity of subsidies to make the service viable. Therefore, it should be noted that our semiflexible DRFS uses roughly the same resources as the traditional feeder systems.

Depending on the transportation system proposed, different levels of flexibility are considered that may affect stops, routes, timetables, or prebooking requirements [9]. On the most limited flexibility level, systems have been proposed with fixed bus stops, routes, and timetables, but which run only when a booking is made. On the other end of the spectrum, a fully flexible system operates without predetermined bus stops, routes, or timetables, serving users in a service resembling that of a taxi. These fully flexible systems typically run using smaller vehicles, and routes are created according to passengers’ requests in a ridesharing format, with fixed or variable fares according to the travelled distance and service availability. Some papers compare these fully flexible systems with other systems or traditional bus systems using simulations [10–12]. Other papers focus on developing guidelines for the operation and classification of these systems [13].

According to a recent and comprehensive survey and classification of demand-responsive public bus systems [14], our DRFS corresponds to a many-to-one, semiflexible, stop-based, static demand responsive bus system. This means it is feeder line (“many-to-one”), with fixed stops (“stop-based”, not “door-to-door”), but flexible routes and timetables.
(“semi-flexible”), which are designed before the operation starts (“static”). Moreover, with the objective of minimizing the passenger travel time, it takes the passengers’ perspective. Only a few of these systems have been considered before [15, 16], but they consider systems with a completely different design.

Still, other semiflexible feeder services, combining characteristics of both the traditional and demand-responsive systems, can be found in the literature. For example, a customized bus (CB) offers a regular operation optimized to passengers’ requests, determining routes according to the pick-up and drop-off locations of passengers. The operation is adjusted on a daily basis, based on the clustering of demand in the service area [17–19]. A demand-responsive connector (DRC) operates a fully flexible door-to-hub service in cycles [20]. However, the cost of such taxi-like or shuttle transit mode requires large cities with dense populations for a profitable operation [21].

The resulting system from merging a demand-responsive and a traditional system can generate a substitute for the traditional system or a complementary system running in parallel. Nourbakhsh and Ouyang [22] explore a flexible service for low-demand areas, offering a door-to-door service that substitutes the traditional operation. This system, however, is applied only for a grid network. In another complementary system for high-demand areas with a grid structure, Uchimura et al. [23] design a hierarchical structure. On the upper level, traditional mass transit lines connect different areas with an express service. The intermediate layer operates an intercommunity traditional transit system offering hub stations where passengers can switch to the last layer, which is the intracommunity service designed as a fully flexible door-to-door demand-responsive system. Only a few of these systems could be used as a replacement system in low demand areas.

Another aspect taken in account for semiflexible services is the way to structure the routes for the buses. Pratelli et al. [24] design a demand-responsive feeder system were part of the stops are fixed, as in the traditional system, but the demand-responsive operation allows deviations to include a limited number of extra requests from optional stops. A methodology to choose the most suitable policy is defined according to the level of demand and the accommodation rate [15]. In our semiflexible DRFS, the structure of the lines from the traditional feeder system are kept, such as a fixed terminal to begin the service and a common destination for all requests. Limited flexibility from this original line allows optimizing the operation in off-peak hours considering the preferred departure time of the passengers.

To conclude, our DRFS operates in suburban areas as a many-to-one, semiflexible, stop-based, static demand-responsive bus system, in order to substitute a traditional system in off-peak hours. Therefore, it keeps the traditional feeder systems’ structure but includes flexibility in order to allow passengers to request a trip to the hub station at their preferred departure time. Mainly, the aspects that it is a semiflexible system and that it operates in off-peak hours making it different compared to other systems in the state of the art.

3. Problem Statement

This section discusses the assumptions on the semiflexible demand-responsive feeder system (DRFS) and the required input data. The starting point is a traditional feeder system (TFS) operating a fixed number of bus lines with standard routes and a regular timetable in a suburban area. This system is optimized for the average daily demand.

The DRFS is designed to replace the TFS during off-peak hours and will modify the standard route for each bus line towards the hub station and the departure time of the scheduled buses based on passenger requests. For every operation period of the problem, a list of passenger requests is available. Each request represents a trip by a passenger, with a desired bus stop and preferred departure time to start the trip. The standard route of each bus line can be modified with one or more shortcuts and limited detours. With “limited detours,” we mean that in each detour at most two additional stops can be visited before returning back to the standard route. This assumption guarantees, on the one hand, that lines are still similar to the standard lines, and on the other hand, that sufficient opportunities are available for shortcuts to skip bus stops without passengers and making detours when appropriate. For the purpose of limiting the flexibility in this study, the value was set to at most two additional stops per detour, creating a balance between a fully flexible DRFS and the TFS with fixed lines. This value, however, could be influenced by several other factors in reality, such as the availability of an alternative route between stops, or the ability to visit multiple different stops with only a small detour. This results in a limited pool of possible routes for each bus line.

Figure 1 presents an example of the DRFS operation. As in the traditional system, a bus line is indicated by a unique nomenclature (a number, a code, or a color, for instance) and this nomenclature is used by passengers to identify the bus line they need to their destination. In Figure 1, the standard line of the TFS is indicated as a solid line, with bus stops 6, 5, 4, 1, and the last stop S. In the DRFS, it can be decided, before the operation starts, to include the dashed detour, picking up passengers at stops 3 and 2 and skipping stops 5 and 4. This result in two possible routes for this line: the standard line and the line with the dashed detour.

3.1. Assumptions for the DRFS. The DRFS runs in a suburban area represented by a graph $G = (V, E)$, where $V$ is the set of bus stops $(i = 0, 1, \ldots, n)$, and $i = 0$ representing the hub station. Edges $E$ indicate links between bus stops $i \in V$ and $j \in V$ at a travel cost of $c_{ij}$. Some initial assumptions for the operation are as follows:

(i) The objective of the DRFS is to minimize the total travel time of the passengers;
(ii) The service operates for $n$ consecutive periods;
(iii) There are a fixed number of lines connecting all the bus stops to the hub station;
(iv) All lines start at a predefined terminal bus stop and head to the hub station;
(v) Each bus is associated with a line, following a route from the pool of routes generated for that line;
(vi) The pool of routes includes all feasible routes for each line, considering stop skipping and limited detours;
(vii) The route length is limited to the duration of the operation period ($P_{op}$);
(viii) The buses depart from the terminal and arrive at the hub station within the operation period;
(ix) The departure times of the buses are not fixed but can be adjusted within the operation period;
(x) The bus capacity is assumed to be sufficient to address all requests.

For each period of operation, there is a list of requests representing passengers that want to use the service during that period. Each request is composed of an origin bus stop and a preferred departure time ($p_r$) representing the moment a passenger will be ready to take the bus towards the hub station. A request can be accepted or rejected. If accepted, it means that this passenger will be picked up by one of the buses in that operation period at pick-up time ($p_p$) and will reach the station at drop-off time ($p_d$). If a request is rejected, it means no bus will pass by the passenger’s bus stop after its preferred departure time, and a penalty will be associated with this request. Rejected requests are then included in the request list for the following operation period. After the last optimized off-peak period, it is assumed that again the TFS is operated, serving all stops. Therefore, all passengers will be served.

The main objective of the DRFS is to minimize the passenger total travel time ($p_{tt}$) by selecting, in each period, the best route and best departure time for each bus. The total travel time consists of the passenger waiting time ($p_{wt}$) at the bus stop and the passenger in-vehicle time ($p_{vt}$) until passengers arrive at the hub station. Therefore, the total travel time for each passenger given in the following equation is the difference between drop-off at the hub station and the preferred departure time:

$$p_{tt} = p_{wt} + p_{st} = (p_{p} - p_r) + (p_d - p_p) = p_d - p_r. \quad (1)$$

In order to determine the penalty for a rejected request, it is assumed that passengers will be waiting for a bus until the next operation period. Since the routes for the next period are not defined yet, and since all buses depart from the terminal and arrive at the station within the operation period, it means that, in the worst case, rejected requests will arrive at the station by the end of the following operation period. Therefore, a passenger penalization ($p_{pen}$) is given by the time difference between the passengers’ preferred departure time and the end of the following operation period given by the following equation:

$$p_{pen} = (p_{op} - p_r) + p_{op} = 2 * p_{op} - p_r. \quad (2)$$

3.2. Generation of the Pool of Routes for the Lines. The problem formulation of the DRFS uses a limited pool of alternative routes for each bus. In this subsection, we develop a method to generate such a pool. The proposed semiflexible DRFS focuses on operating a flexible service but keeping the main characteristics of the TFS. Therefore, we start from the standard (TFS) route of each line and then generate a set of alternative routes using “stop skipping” and “limited detours.”

The standard routes are optimally designed for the TFS, serving all bus stops and efficiently accommodating all the demand over multiple periods. In the DRFS, buses do not follow these standard routes, but decide for the best route according to the requests for each operation period. Stop skipping and limited detours are considered. Stop skipping means that it is possible to skip one or a sequence of bus stops in the standard route. Detours mean that it is possible to include additional bus stops in the standard route. Since the DRFS is not a fully flexible system where buses are allowed to take any possible route, we limit the detours to including at most two other bus stops before returning to a bus stop in the standard line. However, it is possible to combine detours with stop skipping. Furthermore, cycles are not allowed, which means that a bus stop cannot be visited twice by the same bus. Moreover, a limit is set on the maximum route length to control the maximum travel time a bus can ride. It should be noted that overlaps between different lines are possible, resulting in more than one line serving the same bus stop.

By considering the limited detours, it is possible to precalculate all possible routes for each line in the DRFS. These alternative routes are then stored in a pool of routes that are used by the memetic algorithm instead of recalculating all possible movements while exploring solutions for the routes. As an example, Figures 2 and 3 illustrate a TFS operation in a grid network composed of nine bus stops and a station $S$, served by a red line (Figure 2) and a blue line (Figure 3). Figures 2 and 3 also indicate the possible detours or stop skipping for these lines, based on the previously
stated assumptions. For instance, the original route of the blue line includes the sequence of stops 9-8-7-4-1-0. When the bus arrive at the stop number 8, it is possible to include a detour to stop number 5 instead of continuing the initial route to stop 7. From this point, the bus can return to the original line, visiting stop number 4 and skipping the bus stop number 7, or include another detour to visit stop number 2. After this second detour, the bus is obliged to return to the original line. The stop 3 cannot be included in the blue route since it would require a detour of more than two stops to return to the standard route. That is, from stop 9, it would require a detour to stop number 6 and then to stop number 3, but then it would require at least one more stop to return to the original line, which is not allowed.

3.3. List of Requests and Operation Period. Every passenger that wishes to use the system submits a request composed of a bus stop and a preferred departure time. It means passengers are planning to arrive at the given bus stop and will be waiting for the bus from that moment on. Since the off-peak operation is divided in periods of operation, requests have to be submitted before the start of the corresponding operation period. Once the buses of an operation period start running, no new requests for that operation period are allowed. Therefore, the requests considered for the optimization of an operation period are those related to that operation period and unserved requests from previous operation periods.

4. Memetic Algorithm Approach

Optimizing the performance of the DRFS discussed in the previous section is addressed with a memetic algorithm (MA), combining an evolutionary heuristic suitable for exploring large areas of the search space in coordination with repair and improvement functions, acting as local search techniques, to improve solutions to fit the preferred departure time of passengers. The framework of the proposed MA uses different iterations to configure the buses’ operation, i.e., route assignment and departure time for each bus. An overview of the MA is presented in Algorithm 1.

Figure 4 illustrates the flowchart of the MA. For each period of operation, the requests received for that period, the rejected requests from previous periods and the list of possible routes for each line are considered as input. Subsequently, a generation with initial solutions is created and the fitness value for each solution is calculated. Then, the process of selection, crossover, mutation, repair, and improvement is repeated until a given number of generations is produced without further improvement of the best solution.

For each operation period in the operation horizon, a solution is represented by a list of bus stops and a departure time from the terminal for each bus line of the service. The route is represented by a sequence of bus stops and the departure time with a delay from the beginning of the operation period. For instance, a possible representation for the red line service on Figure 2 could be the route that departs from the terminal 20 minutes after the operation period begins and takes a shortcut straight to bus stop 2, where the standard route is followed until the station.

4.1. Population Generation. The MA approach aims to identify, for each operation period, the solution that minimizes the total passenger travel time. The first step of the MA is to populate the initial generation with solutions. Since all buses must arrive at the station before the end of the operation period, the route length determines the maximum delay time a bus can have. Therefore, the route is assigned first; selecting a random route from the pool of routes for each line, and then, a departure time is assigned randomly limited to the maximum delay time. This random generation tries to cover a large number of possible combinations of routes and departure times as possible, which are then subjected to further improvements in the following generations.

4.2. Evaluation of Solutions. After assigning a route and a departure time to each line of the operation period, it is
possible to generate the timetable for all bus stops and the arrival time of the buses at the station. This information is used to evaluate the objective function value of the solution using the list of requests for the current operation period. The fitness of a solution is given by the sum of passenger travel time for all requests in the list of requests \((r \in R)\) as follows:

\[
\text{Objective function value} = \sum_{r \in R} \text{p}_{tt} + \text{p}_{\text{pen}}.
\]

(3)

For every request in the list of requests, the passengers’ preferred departure time \((p_r)\) is compared with the timetable for that bus stop. If a passenger arrives at the bus stop before the bus, the request is accepted and \(p_p\) and \(p_d\) are updated. If the passenger arrives at the bus stop after the bus passed by, or no bus visits the stop in that operation period, the request is penalized, and this request is included in the list of requests for the next operation period. If more than one bus from different lines can serve a passenger in the same operation period, the one that arrives earlier at the station is chosen. It should be noted that this is not necessarily the first bus serving the stop.

After evaluating the new solutions in a generation, they are sorted according to the fitness value, and the solution with the smallest fitness is set as the best solution of the generation. Also, the top 5% of the solutions are set as good solutions for the elitist crossover method (below). If the population size is smaller than 100 solutions, at least five solutions are set as good solutions. The evaluation fitness function is presented in Algorithm 2.

4.3. Crossover Operation. The first operation to populate new generations is the crossover. The MA uses two different methods: an elitist and a random crossover, with a probability of 50% each. In the elitist method, at least one of the
parents is selected from the group of good solutions. The other parent is randomly selected. In the random crossover, both parents are selected randomly from the general population. Once the parents are selected, the crossover selects the routes of the lines and the departure time from the parents to generate two new solutions. It is done by randomly assigning to each line of one new solution, the route and departure time used by a randomly selected parent on this line. The nonselected route (and departure time) for that line is then assigned to the second new solution. The crossover operation is presented in Algorithm 3.

4.4. Mutation Operation. The mutation operation includes variability in the generations by changing routes and departure times of a solution after the crossover method, with a 5% probability of occurring. A mutation can be a small or a large mutation, each with a 50% probability. In the small mutation, one route and departure time of the solution is randomly modified. In the large mutation, all routes and departure times are generated again for that solution, resulting in a completely new solution. These new solutions are introduced to stimulate diversification in the population. After the mutation operation, the new solution is included in the new generation list, whether mutated or not. The mutation operation is presented in Algorithm 4.

4.5. Repair Operation. After populating the new generation, all solutions pass through both the repair and the improvement operations. The repair operation modifies the solution, trying to avoid penalized requests and accepting them in the current period. First, all rejected requests in the list of requests are categorized according to the type of penalty, i.e., whether the penalty results from an early bus service or from an unserved bus stop. An early bus service means a passenger arrives at the requested bus stop after the last bus has already served that bus stop, and consequently, this passenger is not served. In the case of an unserved bus stop, the bus stop is not visited by any bus in the current solution, and passengers requesting travel from these stops are forced to wait for the next period.

Once all penalized requests are categorized, one of them is randomly selected to be “repaired” by forcing this request to be served on one of the lines. If the request is rejected due to early service, the repair function tries to delay the corresponding bus as much as possible, so the bus arrives at the bus stop at the same time as the passenger. It consequently delays all arrival times for all the bus stops in the route and thus increases the waiting times for other passengers. Obviously, this delay is limited by the fact that the bus should arrive at the hub station before the end of the operation period.

A more complex procedure is executed if the request is rejected due to an unserved bus stop. First, one of the possible lines that may serve the bus stop is selected. In the example of Figures 2 and 3, for instance, bus stop three can only be served by the red line, bus stop seven only by the blue line, and all the others bus stops by both lines. Second, considering the current route for the selected line, all bus stops where passengers are being served by this route are marked, and a new route including all these marked bus stops and the bus stop of the rejected request is chosen from the pool of routes for this line. Finally, the procedure checks whether the departure time needs to be adjusted, either because the new route is longer than the previous or because the arrival time at the included bus stop is still not serving the penalized request.

An example of this procedure is illustrated in Figure 5. In this example, there is a penalized request at bus stop four, not served by neither the red line (a) nor the blue line (b), since this bus stop can be included in both the lines, stops with served passengers on the current route of the lines are marked: stops 8 and 1, if the red line is chosen, and stops 8 and 2 for the blue line. There are no possible routes including the stops 8, 4, and 1 in the same route for the red line, therefore, this option is discarded. For the blue line, it is possible to change the route to the sequence illustrated on Figure 5(c). The last check adjusts the departure time for the optimized request. The bus in the blue line departs at the beginning of the operation period. It will serve bus stop 4, 15 minutes later with the new route. If the passenger arrives any time earlier, there will be no adjustment in the departure time of the bus. If the passenger arrives later, the departure time of the bus will be delayed as much as possible, so this passenger does not have to wait for the bus, respecting the maximum arrival time at the station.

```plaintext
ALGORITHM 2: Fitness evaluation.

Start:
Solution fitness = 0
For request in list of requests:
   If preferred departure time at request stop ≤ bus arrival at request stop:
      \[ P_P = \text{time bus arrive at bus stop} \]
      \[ P_d = \text{time bus arrive at station} \]
      \[ P_{\text{t}} = P_d - P_r \]
      Solution fitness += P_{\text{t}}
   Else:
      \[ P_{\text{pen}} = 2 \times P_{\text{op}} - P_r \]
      Solution fitness += P_{\text{pen}}
Return Solution fitness
```

"
Start:
Random selection of crossover method: elitist or random
If elitist:
Choose random parent \( P_1 \) from the list of best solutions
If random:
Choose random parent \( P_1 \) from previous generation
Choose second random parent \( P_2 \) from previous generation
Create empty offspring \( \rightarrow O_1, O_2 \)
For line in set of lines:
Random selection order: true or false
If true:
\( O_1 \) receives line configuration from \( P_1 \)
\( O_2 \) receives line configuration from \( P_2 \)
If false:
\( O_1 \) receives line configuration from \( P_2 \)
\( O_2 \) receives line configuration from \( P_1 \)
Return \( O_1, O_2 \)

Algorithm 3: Crossover operation.

Start:
mutation probability = Random number between (0, 100)
If mutation probability \( \leq 5 \):
Choose mutation size: small or large
If small:
Choose a random line from solution
Select a random route in the pool of routes for this line
max departure time = operation period time–route length
bus departure = random time between (0, max departure time)
If large:
For each line in set of lines:
Select a random route in the pool of routes for this line
max departure time = operation period time–route length
bus departure = random time between (0, max departure time)
Return Solution

Algorithm 4: Mutation operation for new solutions.

Figure 5: Example of a repair operation for a small grid network to include a request in bus stop 4 in the red or the blue line: (a) the current red line with passengers at stops 8 and 1; (b) the current blue line with passengers at stops 8 and 2; (c) final solution adapting the blue line.
After finishing the repair operation, the new solution is re-evaluated, and if the fitness is improved compared to the previous one, the new repaired solution replaces the earlier solution in the new generation. Otherwise, it is discarded.

4.6. Improvement Operation. The last operation included in the MA is the improvement operation. While the repair operation tries to delay the departure time of buses or to include stops not served in the routes, the improvement operation advances the departure time of a bus or makes the route more direct to the destination. The operation starts with the selection of a random request from the list of accepted requests. After checking which bus is currently serving the request, two steps are executed. First, it tries to find a new route aiming to reduce the in-vehicle time for the passenger of this request. As in the previous operation, all bus stops serving passengers along the current route are marked. Then, a new route is selected for this line only when there is a shorter route for the selected request with all the marked bus stops along the route. Second, the bus departure is set earlier as much as possible to reduce the waiting time for this request. The new solution replaces the earlier one in the new generation if the improvement operation improves fitness.

The operation of the improvement function is illustrated in Figure 6. In this example, there is a request for bus stop 2 with the preferred departure time 30 minutes after the period starts, and this passenger is served with a bus from the blue line following route illustrated on the left in Figure 6. The current departure time from the terminal is 25 minutes after the beginning of the period, and the bus is picking up the selected passenger at time 40. Therefore, the waiting time for this passenger is 10 minutes. In the improved solution, on the right in Figure 6, the route is kept unchanged since there is no route in the pool of routes with a shorter in-vehicle time from bus stop 2 to the station. Then, the bus is set to depart 10 minutes earlier, consequently arriving earlier at every bus stop along the way and picking up the passenger at bus stop 2 without any waiting time. If other passengers are being served by this bus, there are two options: their waiting times are reduced by 10 minutes, or their requests are rejected in this solution. After updating the route and the timetable, the new solution is re-evaluated, and if the fitness is better, the new solution replaces the earlier one in the new generation. Otherwise, it is discarded.

5. Results

In this section, results for the MA are presented in six analyses: First, benchmark instances for the DRFS are created and the parameters of the MA are evaluated in a sensitivity analysis. The MA is then validated through a comparison with a mathematical model that solves a very similar but somewhat easier problem to optimality. Second, the MA solutions for the benchmark instances are compared with those of a TFS. Third, instances are generated with variations in the number of lines. Fourth, more complex instances without restriction on the possible detours are created. In these instances, buses are allowed to take any route in the network from terminal to station. Finally, the MA is evaluated on larger instances, with 70 bus stops in the area instead of 25 bus stops. These two last experiments allow evaluating the viability of the MA to find local optima for complex instances and to check the performance of the DRFS.

5.1. Network and Instances. Since this DRFS was not studied before, there are no benchmark instances available to evaluate the performance of the MA. Therefore, new instances are created based on a graph corresponding to a suburban area served with a feeder bus system. It is composed of 25 bus stops connected to a transfer station (S) and served by three lines. All lines start at bus stop/terminal 25. Each bus stop expects 1 to 6 passengers, with a total of 100 requests over five consecutive operation periods. The time a bus takes from one bus stop to another is equivalent to the Euclidian distance between the bus stops, converted to minutes, and presented in Figure 7. The length for the standard lines for the TFS and the number of alternative routes in the pool of routes for each line in the DRFS are presented in Table 1. The lines for the TFS are optimized to serve the total demand presented in Table 2. These lines are obtained with an exact optimization approach developed for the bus line planning problem considering the average daily demand expected at the different bus stops [25]. The objective is to minimize the passenger travel time from the different bus stops to the hub station. It results in the optimal standard routes for the lines of the TFS. In a second step, the frequency of the service is set to one bus per line per operation period. This procedure is similar to the planning of traditional feeder systems in practice.

One instance with five operation periods is shown in Table 3. In each operation period of this example, requests are concentrated in 9 to 15 different bus stops of the 25, corresponding to, on average, half of the bus stops per operation period. This means that in the total operation horizon, there are 50% of stops without requests (50% of empty bus stops). Several instances with random distributions of demand were created, categorized according to the number of empty bus stops in the total operation: 30%, 50%, or 70%.

5.2. Sensitivity Analyses for MA’s Parameters. This sensitivity analysis examines the parameters for the MA, i.e., the size of the population in each generation and the stop criteria in the search for improvements. In order to somewhat limit the computation time required to optimize an operation period, population sizes of 50, 100, or 200 are considered combined with a maximum number of generations without improvements of 20, 50, or 100. Beyond these parameters, the total number of generations was limited to 1,000. During the experiments reported in this paper, however, this value was never reached.

The results for this sensitivity analysis are presented in Table 4 for 15 instances grouped in three categories according to the percentage of empty stops (30%, 50%, and
70%), with five instances for each group. The results present the average passenger travel time in minutes, and the computation time in seconds. Considering the limited available time for computations just before each operation period, and considering the quality of the results, we conclude that a population size of 100 solutions and a stop criteria of 50 generations without improvements are appropriate parameter values, increasing the population size from 100 to 200 or the maximum number of generations without improvement from 50 to 100 produced the same passenger travel time on average (31.8 min) for these 15 instances, but the computation time increased from 55 to 126 and 132 seconds, respectively. Reducing these parameters increased the optimal value of the solutions. We conclude that the selected parameters values guarantee (at least for these instances) a reasonable number of generations before reaching the stop criterion and sufficient diversity in the population.

<table>
<thead>
<tr>
<th>Lines</th>
<th>Route lengths (min)</th>
<th>Pool of routes size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>32.1</td>
<td>192</td>
</tr>
<tr>
<td>Green</td>
<td>33.2</td>
<td>60</td>
</tr>
<tr>
<td>Blue</td>
<td>28.2</td>
<td>225</td>
</tr>
</tbody>
</table>

| Combinations of routes: | 2,592,000 |
| Operation period duration: | 60 min |

Table 1: Network characteristics: length of lines for the standard route in the TFS; number of routes in the pool of routes for each line in the operation period.

Figure 6: Illustration of improvement operation for a request and the improved line, advancing the departure time of the bus 10 minutes to reduce the waiting time of this passenger.

Figure 7: Network representing a suburban area, served by three lines in a TFS.

Table 2: Expected demand for each bus stop during the entire operation horizon.

<table>
<thead>
<tr>
<th>Stops</th>
<th>Demands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>6</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>21</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>25</td>
<td>4</td>
</tr>
</tbody>
</table>
5.3. Performance of the MA. This first analysis evaluates the performance of the MA by comparing its solutions with the optimal solutions obtained with the exact model of a similar problem solved with CPLEX. In this model, buses depart at the beginning of the operation period for all lines, and no exact arrival times of passengers at the bus stops are considered. Instead, all passengers are assumed to be present at the bus stop at the beginning of each operation period. Therefore, in all instances considered in this subsection, all preferred departure times are set equal to the beginning of the operation period. Due to this simplification, the model can be solved to optimality for the size of instances considered. The model is presented and discussed in Appendix A (available here).

For this analysis, data for twenty operation horizons were randomly generated for each percentage of empty stops, and with five operation periods each. Therefore, 300 operation periods are optimized both with the mathematical model in CPLEX and using the MA optimization for the DRFS. The routes for the demand illustrated in Table 3 are shown in Figure 8 as an example for the three solutions (TFS and optimal solution and MA solution for the DRFS) on all five operation periods. Buses depart at the beginning of the operation period for all lines in the TFS and in the optimal solution for the DRFS, due to the initial assumptions. In the solution found by the MA for the DRFS, the departure times identified by the algorithm were also equal to the beginning of the operation period for all lines.

For this analysis, data for twenty operation horizons were randomly generated for each percentage of empty stops, and with five operation periods each. Therefore, 300 operation periods are optimized both with the mathematical model in CPLEX and using the MA. Both waiting and in-vehicle times of passengers are evaluated, considering the standard routes for the TFS, and the optimal routes obtained by CPLEX, and the routes obtained with the MA.
<table>
<thead>
<tr>
<th>TFS</th>
<th>DRFS – Mathematical model</th>
<th>DRFS – Memetic Algorithm</th>
</tr>
</thead>
</table>

![Graphs](image-url)

**Figure 8:** Routes for total operation in the TFS (1st column), the optimal solution obtained with the mathematical model for the DRFS (2nd column), and the solution obtained for the DRFS with the MA (3rd column) for all the five operation periods of one instance.
assumptions. In the solution found by the MA for the DRFS, the departure times identified by the algorithm were also equal to the beginning of the operation period for all lines. As mentioned before, routes are identical for all operation periods in the TFS. Six routes in the optimal DRFS solution are the same as in the TFS. It happens in the first three operation periods for the red line and in the first, second, and fifth period for the green line. Eleven out of fifteen routes for the lines in the DRFS obtained with the MA are identical to the optimal DRFS solution. All routes obtained with the MA solution were identical to the optimal solution for the first and fifth operation periods. There are slight variations in the routes obtained for the second and the fourth period. The average passenger travel time for this instance is presented in Table 5 for each operation period. For the TFS, the average passenger travel time was 31.5 min. For the DRFS, this value was 10.8% lower with the optimal solution, representing a reduction of 3.4 minutes. For the solution obtained with the MA, the average passenger travel time was 0.1 minutes longer than the optimal solution.

The optimal travel time for passengers was achieved in four out of five optimized periods with the MA. As mentioned before, the MA found identical routes as in the optimal solution for the first and the last operation periods, so both simulations present the same results. For the second and the fourth operation periods, even though routes were not the same for all the lines as in the optimal solution, the obtained results with the MA were the same as the optimal result. It means that different routes serve passengers with the same efficiency as in the optimal solutions. Passengers in the third period had an average travel time of 29 minutes with the solution of the MA, a value 2.5% above the optimal solution.

This same analysis was repeated for all 60 instances, comparing the average passenger travel time for the TFS, and the optimal and the MA solution for the DRFS (Table 6). In the optimal solutions, the average passenger travel time was 8.6% shorter than the TFS solution, while the MA could improve the passenger travel time by 8.0%. Moreover, the results of the MA were the same as in the optimal solution in 69.3% of the operation periods. Increasing the percentage of empty stops does not make any difference in passenger travel time on TFS because routes for the lines are identical for the whole operation period. In the optimal and the MA solutions for the DRFS, however, increasing the percentage of empty bus stops in the instances allows the possibility to skip empty bus stops and serve busy bus stops faster. With 30% empty stops, passenger travel time was 1.3 minutes shorter in the optimal solutions than in the TFS. With 70% of empty stops, this value was 4.5 minutes shorter. On average, passengers could arrive at the station 2.7 min earlier if the routes of the TFS could be adapted and changed dynamically. Comparing computation times for these simplified instances, the mathematical models took 6.5 seconds on average to identify the optimal solution for each period of operation, and the MA took 52 seconds on average.

In the previous analysis, passengers requested their trips to start exactly at the beginning of the operation period, in order to generate a simplified problem, allowing the comparison with results obtained by solving the mathematical model. In this second part of the analysis, the demand distribution over the different stops is the same as before for the 60 instances, but the preferred departure time can take any value within the operation period. This means that some preferred departure times occur after the bus has already served the respective bus stop, and these passengers are left to be served in the following operation period. This results in a variation in the average passenger waiting time, even for the TFS. For the demand presented in Table 3, the MA solution for the third operation period is illustrated in Figure 9, as an example. For this solution, the bus in the blue line departs two minutes after the operation period begins, at time 02:02, and arrives at the station at 02:35. The bus in the green line departs at 02:17, and arrives at the station at 02:55. The red line bus departs at 02:19, arriving at 02:53, 7 minutes before 03:00, the end of the period. Buses pick up passengers in the bus stops marked in the figure. Results for this instance and routes for the five operation periods in the MA solution are presented in Appendix B, together with the full table with the scheduled departure time for each passenger for this operation period.

5.4. Results of the MA for General Instances. The results for the MA are compared to the TFS for all the 60 instances, grouped by the percentage of empty stops, and presented in Table 7. This table shows the average passenger waiting, in-vehicle, and total travel time for the TFS and the DRFS with the MA solution. On average, the time passengers spend on the bus in the optimal routes of the DRFS is slightly increased compared to the time passengers took while traveling using standard routes in the TFS. However, this value decreases the more empty stops are present in the instances. It means that detours included in the lines’ routes to serve passengers increase the average in-vehicle time, but skipping empty bus stops can reduce it. On the other hand, the average passenger waiting time on the DRFS reduced by 48.1% compared to the TFS, with an overall improvement of 30.6% on passenger travel time. However, since the memetic algorithm’s objective is to improve passenger travel times, the routes taken by buses while not carrying passengers are not always the shortest route possible. This does not imply that these solutions can be further improved, because bus driving time is not part of the objective function and the buses are optimally serving passengers. The average calculation time to achieve the termination criteria for the MA was 56 seconds per operation period.

5.5. Results of the MA for Varying Numbers of Lines. The previous analyses consisted of a network with three lines. In this section, the same network with the same demand is now served by a system with two and four lines instead of three, in order to analyze the impact of that parameter. To make a fair comparison with the TFS, the lines of the traditional feeder system are optimized using the MA, considering the total demand for entire operation horizon both for two and four lines. The standard lines for the TFS services for two and four lines are presented in Appendix C.
The average lengths of the TFS lines with two, three, and four lines are 50.5, 31.2, and 29.3 minutes. The reason why the average length of the lines are roughly the same for three and four lines is that the area considered can be covered easily with only three lines and therefore all three (or four) lines have a length close to the minimal travel time between the terminal and the hub station. Therefore passengers have to wait longer for their buses and spend more time inside the vehicles before reaching the hub station. These results are presented in the first columns of Table 8.

When there are four buses in the semiflexible DRFS, the optimization adjusts better to the passengers’ expectations. On average, the passenger travel time in the DRFS is 37.6% shorter compared to the TFS, while this was 30.6% for three lines and 29.8% for two lines on the DRFS. The passenger waiting time reduced on average with 8.9, 14.3, and 16.8 minutes for two, three and four lines. On the other hand, it is possible to identify a reduction in the in-vehicle times of passengers for two lines only (7.9 minutes).

Table 5: Average $p_{tt}$ for each period of the example simulated.

<table>
<thead>
<tr>
<th>Periods</th>
<th>TFS solution (min)</th>
<th>Optimal solution (min)</th>
<th>MA solution (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>31</td>
<td>28.3</td>
<td>28.3</td>
</tr>
<tr>
<td>2nd</td>
<td>30.9</td>
<td>29.4</td>
<td>29.4</td>
</tr>
<tr>
<td>3rd</td>
<td>31.8</td>
<td>28.3</td>
<td>29</td>
</tr>
<tr>
<td>4th</td>
<td>32.4</td>
<td>27.1</td>
<td>27.1</td>
</tr>
<tr>
<td>5th</td>
<td>31.3</td>
<td>27.4</td>
<td>27.4</td>
</tr>
<tr>
<td>Average</td>
<td>31.5</td>
<td>28.1</td>
<td>28.2</td>
</tr>
</tbody>
</table>

Table 6: Average passenger travel time in the TFS, with optimal routes and the MA. Improvement of optimal and MA solution compared to TFS, classified according to the percentage of empty bus stops in the instance.

<table>
<thead>
<tr>
<th></th>
<th>TFS solution (min)</th>
<th>Optimal solution (min)</th>
<th>Optimal solution improvement (%)</th>
<th>MA solution (min)</th>
<th>MA solution improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30% empty stops</td>
<td>31.5</td>
<td>30.2</td>
<td>4.2</td>
<td>30.5</td>
<td>3.2</td>
</tr>
<tr>
<td>50% empty stops</td>
<td>31.5</td>
<td>29.1</td>
<td>7.6</td>
<td>29.3</td>
<td>7.1</td>
</tr>
<tr>
<td>70% empty stops</td>
<td>31.5</td>
<td>27.0</td>
<td>14.1</td>
<td>27.2</td>
<td>13.7</td>
</tr>
<tr>
<td>Average</td>
<td>31.5</td>
<td>28.8</td>
<td>8.6</td>
<td>29.0</td>
<td>8.0</td>
</tr>
</tbody>
</table>

Table 7: Results of average passenger waiting, in-vehicle, and total travel times for the TFS and the MA grouped by percentage of empty stops in the instances.

<table>
<thead>
<tr>
<th></th>
<th>TFS solution</th>
<th>MA solution</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p_{wtt}$ $p_{vtt}$ $p_{tt}$</td>
<td>$p_{wtt}$ $p_{vtt}$ $p_{tt}$</td>
<td>$p_{wtt}$ $p_{vtt}$ $p_{tt}$ (%)</td>
</tr>
<tr>
<td>30% empty stops</td>
<td>29.1 16.5 15.9 16.9 32.8</td>
<td>29.4 16.5 15.7 16.6 32.3</td>
<td>28.20</td>
</tr>
<tr>
<td>50% empty stops</td>
<td>29.4 16.5 15.9 15.7 16.6 32.3</td>
<td>30.5 16.5 14.7 16.4 31.1</td>
<td>29.50</td>
</tr>
<tr>
<td>70% empty stops</td>
<td>30.5 16.5 14.7 16.4 31.1</td>
<td>30.5 16.5 14.7 16.4 31.1</td>
<td>34.00</td>
</tr>
<tr>
<td>Average</td>
<td>29.7 16.5 15.7 16.6 32.1</td>
<td>30.6 16.5 15.4 16.6 32.1</td>
<td>30.60</td>
</tr>
</tbody>
</table>

The average lengths of the TFS lines with two, three, and four lines are 50.5, 31.2, and 29.3 minutes. The reason why the average length of the lines are roughly the same for three and four lines is that the area considered can be covered easily with only three lines and therefore all three (or four) lines have a length close to the minimal travel time between the terminal and the hub station. With only two lines, longer lines are required to cover the area. Therefore passengers have to wait longer for their buses and spend more time inside the vehicles before reaching the hub station. These results are presented in the first columns of Table 8.

When there are four buses in the semiflexible DRFS, the optimization adjusts better to the passengers’ expectations. On average, the passenger travel time in the DRFS is 37.6% shorter compared to the TFS, while this was 30.6% for three lines and 29.8% for two lines on the DRFS. The passenger waiting time reduced on average with 8.9, 14.3, and 16.8 minutes for two, three and four lines. On the other hand, it is possible to identify a reduction in the in-vehicle times of passengers for two lines only (7.9 minutes). For three and four lines, the passenger in-vehicle times increased.
0.1 minutes. It means that more options for stop skipping to shorten the routes are available with two lines. With three or four lines, detours optimize the passenger waiting time, but the average length of routes are roughly the same.

The average computation time per operation period was 53 seconds for the network with 4 lines, while in the network with two lines, the computation time was 93 seconds. Longer routes also mean a bigger search space for the MA, reflected in the number of routes in the pool of alternative routes.

5.6. Instances with Unrestricted Detours. So far, the possible detours that a bus could make were restricted to at most two bus stops outside the original route. In this analysis, the same instances are considered, but buses are allowed to take any possible route in the network. This will allow to further analyze the performance of the MA and to evaluate the opportunity of allowing more flexibility in the routes. Still, no cycles are allowed, and the route length is limited by the duration of the operation period.

The number of possible routes generated for the pool of routes increases from 192 for the red line, 60 for the green, and 225 for the blue line with at most two bus stops outside the original route (Table 1), to 2,149 routes with unrestricted detours for each line. All possible combinations from the terminal to the station are considered. Therefore, the search space increases from 2.6 million possible combinations to 10 billion. Beyond that, is still necessary to identify the optimal departure time for the buses considering preferred departure time for the requests.

Results for the same 60 instances as in the analysis in Section 5.4 are presented in Table 9, comparing the solutions obtained with the MA for the limited and unrestricted detours. Beyond the initial reduction in travel time observed for the DRFS in comparison to the TFS, the value was further reduced by 1.7% with unrestricted detours. The average passenger waiting time reduced from 15.4 min with limited detours to 13.7 min, 11% shorter. The in-vehicle time was a bit longer, on average 16.6 min per passenger compared to 17.9 min with the limited detours (7.8% longer). An operation period example for this unrestricted detour analysis is presented in Appendix D. As before, increasing the percentage of empty bus stops in the simulation reduced the in-vehicle time as well, and this gain is reflected in the average passenger total travel time. From 32.1 min with limited detours, the average travel time obtained with the MA solution with unrestricted detours was reduced to 31.5 min. The average computation time for the optimizations was 407 seconds.

5.7. Large Network Instances. One last analysis considers a larger network with 70 bus stops and a centralized hub, and 300 requests in five operation periods. The large network and the standard lines of the TFS are presented in Appendix C. Here, again three lines are considered to serve the bus stops, with fixed terminal on bus stops 1, 63, and 70. In order to identify the standard routes for the TFS, the MA was used with the total demand for the entire operation horizon and unrestricted detours for the lines were allowed. This procedure does not guarantee the optimal routes for the lines as in the previous network but leads to a good solution given the complexity of the network. Since routes are now much longer and the same numbers of buses are available, the operation period considers a frequency of one bus every two hours. Therefore, for the DRFS, the routes were limited to not exceed 120 minutes. All the other rules were kept as initially, considering the possibility of deviating from the standard route to include at most two extra bus stops or to skip stops. Fifteen instances were optimized with this network, five for each group according to the percentage of empty stops (30%, 50%, and 70%). As before, the variation between instances occurs on the randomization of the preferred departure time of passengers and the operation period on which the trip is requested. Results are presented in Table 10.

Since the operation period is two hours and requests arrive randomly in the instances, the average passenger waiting time in the TFS is 60.1 min. For this configuration of the network and instances, the MA could be able to reduce this waiting time by 21% to 47.2 min. The in-vehicle time was reduced by 13%, from 41.0 min on the TFS to 35.5 min on the DRFS. The average computation time for this network was 459 seconds. The gain on this network was smaller than

| Table 8: Average passenger waiting, in-vehicle, and travel time for 2 or 4 lines. |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                                 | TFS solution    | MA solution     | Improvement     |                 |                 |                 |
|                                 | $p_{wt}$        | $p_{vt}$        | $p_{tt}$        | $p_{wt}$        | $p_{vt}$        | $p_{tt}$        |
| 2 lines                         |                 |                 |                 |                 |                 |                 |
| 30% empty stops                 | 28.1            | 27.7            | 55.8            | 20.3            | 20.7            | 40.9            | 26.60           |
| 50% empty stops                 | 28.4            | 27.7            | 56.1            | 20.2            | 19.8            | 40              | 28.60           |
| 70% empty stops                 | 28.8            | 27.7            | 56.5            | 18.1            | 19              | 37.1            | 34.30           |
| Average                         | 28.4            | 27.7            | 56.1            | 19.5            | 19.8            | 39.4            | 29.80           |
| 3 lines                         |                 |                 |                 |                 |                 |                 |
| 30% empty stops                 | 29.1            | 16.5            | 45.6            | 15.9            | 16.9            | 32.8            | 28.20           |
| 50% empty stops                 | 29.4            | 16.5            | 45.9            | 15.7            | 16.6            | 32.4            | 29.50           |
| 70% empty stops                 | 30.5            | 16.5            | 47              | 14.7            | 16.4            | 31              | 34.00           |
| Average                         | 29.7            | 16.5            | 46.2            | 15.4            | 16.6            | 32.1            | 30.60           |
| 4 lines                         |                 |                 |                 |                 |                 |                 |
| 30% empty stops                 | 28.4            | 15.5            | 44              | 12.9            | 15.8            | 28.7            | 34.70           |
| 50% empty stops                 | 29.1            | 15.5            | 44.7            | 12.1            | 15.7            | 27.9            | 37.60           |
| 70% empty stops                 | 29.8            | 15.5            | 45.3            | 11.7            | 15.3            | 27              | 40.40           |
| Average                         | 29.1            | 15.5            | 44.7            | 12.3            | 15.6            | 27.9            | 37.60           |
When comparing the performance of the DRFS, optimized with the MA, to the TFS for a network composed of 25 bus stops and three lines, the DRFS reduced the average passenger waiting time by 48.8% with the limited detour policy, and 53.5% with the unrestricted detour policy. However, the computation time increased from 56 to 459 seconds to optimize one operation period. These results prove that a semiflexible DRFS allowing limited detours can provide a good solution in a faster time than a fully flexible DRFS. Despite somewhat long computation times, typical for evolutionary algorithms that perform a random exploration of the search space, these results also confirm the good performance of the MA.

When evaluating the same instances with the DRFS obtained by the MA, and comparing it with the results of the same instances in the TFS, the average passenger waiting time reduced with 31.4%, 48.1%, and 57.7% for the same network with two, three, and four lines, respectively. Despite slightly increasing the passengers’ in-vehicle time with 0.6%, 0.6%, and 8.4%, respectively, the total travel time for the DRFS is always smaller than in the TFS. This reduction varies between 6% and 73%.

Another characteristic of off-peak operations, which are the focus of this DRFS, is that there are a lot of empty bus stops along the routes. This feature is explored in these experiments by categorizing the instances according to the percentage of empty stops during the operation horizon. Increasing the percentage of empty stops in an instance, from 30% over 50% to 70% increased the benefits of the DRFS. For the same network, the average passenger travel time reduced with 28.2%, 29.6%, and 34.0% for these clusters, respectively. In the network with 70 bus stops and three lines, the average passenger travel time reduced with 15.6%, 17.0%, and 22.0%, respectively. These results confirm the efficiency of the DRFS in low-demand off-peak operations. The overall gains, however, are dependent on the network configuration and the line characteristics, as well as the potential for shortcuts.

The results presented in this paper demonstrated the benefits of operating a DRFS to substitute a TFS. As such, it could be used to improve efficiency of transit system in suburban areas. However, the presented DRFS could be further improved. For instance, by allowing buses to arrive at the station or depart from the terminal outside the operation period and by cancelling services without passengers, the first situation can also be observed in the results in Appendix B, where requests arriving late in the period are served during the following period only. Allowing buses to arrive at the hub station after the operation period has finished could improve the operation and increase the responsiveness of the system. However, it also means these buses might not be available on time for the next operation period. This future implementation, however, would change the characteristic of the system from static to more dynamic, with a rolling horizon operation period and the possibility to reoptimize the system when new requests arrive and some services are already in operation. This approach is considered as future research.

Also, as mentioned in the results in Section 5.4, the bus routes are not optimized when the buses are not carrying passengers, which might lead to unnecessary detours. Therefore, an improvement could be to modify the objective function value to remove these needless detours. It means

---

**Table 9:** Average passenger waiting, in-vehicle, and travel time for 60 instances comparing limited detour policy and unrestricted movements for the lines.

<table>
<thead>
<tr>
<th></th>
<th>2 stops detour</th>
<th>Unrestricted</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P_{wt}$</td>
<td>$P_{vt}$</td>
<td>$P_{tt}$</td>
</tr>
<tr>
<td>30% empty stops</td>
<td>15.9</td>
<td>16.9</td>
<td>32.8</td>
</tr>
<tr>
<td>50% empty stops</td>
<td>15.7</td>
<td>16.6</td>
<td>32.4</td>
</tr>
<tr>
<td>70% empty stops</td>
<td>14.7</td>
<td>16.4</td>
<td>31.2</td>
</tr>
<tr>
<td>Average</td>
<td>15.4</td>
<td>16.6</td>
<td>32.1</td>
</tr>
</tbody>
</table>

---

**Table 10:** Average passenger waiting, in-vehicle, and travel times for TFS, and the solution of the MA for the DRFS considering the large network.

<table>
<thead>
<tr>
<th></th>
<th>TFS solution</th>
<th>MA solution</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P_{wt}$</td>
<td>$P_{vt}$</td>
<td>$P_{tt}$</td>
</tr>
<tr>
<td>30% empty stops</td>
<td>59.5</td>
<td>41.0</td>
<td>100.5</td>
</tr>
<tr>
<td>50% empty stops</td>
<td>60.4</td>
<td>41.0</td>
<td>101.4</td>
</tr>
<tr>
<td>70% empty stops</td>
<td>60.4</td>
<td>41.0</td>
<td>101.3</td>
</tr>
<tr>
<td>Average</td>
<td>60.1</td>
<td>41.0</td>
<td>101.1</td>
</tr>
</tbody>
</table>

---

The previous network, where lines could better “assist” other lines by taking over bus stops. In this larger network, stop skipping is not always used due to the unavailability of other lines to take passengers from the skipped stops. Still, significant improvements of on average 18.2% were obtained.

### 6. Conclusion and Future Research

This paper presents a semiflexible demand-responsive feeder system (DRFS) as a substitute to a traditional feeder system in suburban area. The semiflexible DRFS runs a service similar to a traditional feeder system (TFS), but with adjustable routes and timetables according to passengers’ requests. Its objective is to optimize passenger travel time. From the operators’ perspective, the proposed system allows to offer a more flexible system to the passengers using the same resources. It does require a reservation system where passengers can make requests.

In order to optimize the operations of the DRFS, a memetic algorithm (MA) is developed, merging an evolutionary metaheuristic with local search operations. The results of the MA were found to have a travel time of 0.1 minute longer than the optimal solution. Moreover, the results of the MA were the same as in the optimal solution for seven out of ten periods.

When comparing the performance of the DRFS, optimized with the MA, to the TFS for a network composed of 25 bus stops and three lines, the DRFS reduced the average passenger waiting time by 48.8% with the limited detour policy, and 53.5% with the unrestricted detour policy. However, the computation time increased from 56 to 459 seconds to optimize one operation period. These results prove that a semiflexible DRFS allowing limited detours can provide a good solution in a faster time than a fully flexible DRFS. Despite somewhat long computation times, typical for evolutionary algorithms that perform a random exploration of the search space, these results also confirm the good performance of the MA.

When evaluating the same instances with the DRFS obtained by the MA, and comparing it with the results of the same instances in the TFS, the average passenger waiting time reduced with 31.4%, 48.1%, and 57.7% for the same network with two, three, and four lines, respectively. Despite slightly increasing the passengers’ in-vehicle time with 0.6%, 0.6%, and 8.4%, respectively, the total travel time for the DRFS is always smaller than in the TFS. This reduction varies between 6% and 73%.

Another characteristic of off-peak operations, which are the focus of this DRFS, is that there are a lot of empty bus stops along the routes. This feature is explored in these experiments by categorizing the instances according to the percentage of empty stops during the operation horizon. Increasing the percentage of empty stops in an instance, from 30% over 50% to 70% increased the benefits of the DRFS. For the same network, the average passenger travel time reduced with 28.2%, 29.6%, and 34.0% for these clusters, respectively. In the network with 70 bus stops and three lines, the average passenger travel time reduced with 15.6%, 17.0%, and 22.0%, respectively. These results confirm the efficiency of the DRFS in low-demand off-peak operations. The overall gains, however, are dependent on the network configuration and the line characteristics, as well as the potential for shortcuts.

The results presented in this paper demonstrated the benefits of operating a DRFS to substitute a TFS. As such, it could be used to improve efficiency of transit system in suburban areas. However, the presented DRFS could be further improved. For instance, by allowing buses to arrive at the station or depart from the terminal outside the operation period and by cancelling services without passengers, the first situation can also be observed in the results in Appendix B, where requests arriving late in the period are served during the following period only. Allowing buses to arrive at the hub station after the operation period has finished could improve the operation and increase the responsiveness of the system. However, it also means these buses might not be available on time for the next operation period. This future implementation, however, would change the characteristic of the system from static to more dynamic, with a rolling horizon operation period and the possibility to reoptimize the system when new requests arrive and some services are already in operation. This approach is considered as future research.

Also, as mentioned in the results in Section 5.4, the bus routes are not optimized when the buses are not carrying passengers, which might lead to unnecessary detours. Therefore, an improvement could be to modify the objective function value to remove these needless detours. It means
optimizing the operation from the passengers’ perspective and from the operator’s perspective, aiming to offer an efficient operation in terms of vehicle kilometers as well. Another limitation of the current DRFS is that it considers only the trips towards the hub station. Transfer times for passengers at the hub station are not considered, assuming a high frequency service from this point. In practice, hub stations concentrate several different line services, and transfer time coordination plays a crucial role to improve passengers’ experience using public transport. Synchronizing these services at the station, as covered by Barabino et al. [26], and considering the transfer time at the station, as well as implementing a bi-direction service could improve the service offered by the semiflexible DRFS and increase the success of such a system.

Data Availability

All instances and detailed results are made available at https://www.mech.kuleuven.be/en/cib/drbp/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This project was supported by Internal Funds KU Leuven (Project no. C24M/19/055).

Supplementary Materials

Appendix A: a mathematical model of a (simplified) semiflexible Demand-Responsive Feeder System. Appendix B: an example result for one instance and discussion. Appendix C: layout of the network used for the instances and the traditional fixed lines. Appendix D: an example solution of the MA with unrestricted routes. (Supplementary Materials)

References

[20] X. Li and L. Quadrifoglio, “Feeder transit services: choosing between fixed and demand responsive policy,” Transportation